

# CS 224S LING 281 Speech Recognition and Synthesis

Lecture 15: Dialogue and Conversational Agents (III)  
Dan Jurafsky

CS 224S W2006

## New Idea: Modeling a dialogue system as a probabilistic agent

- A conversational agent can be characterized by:
  - The current knowledge of the system
    - A set of states  $S$  the agent can be in
  - a set of actions  $A$  the agent can take
  - A goal  $G$ , which implies
    - A success metric that tells us how well the agent achieved its goal
    - A way of using this metric to create a strategy or policy  $\pi$  for what action to take in any particular state.

CS 224S W2006

## What do we mean by actions $A$ and policies $\pi$ ?

- Kinds of decisions a conversational agent needs to make:
  - When should I ground/confirm/reject/ask for clarification on what the user just said?
  - When should I ask a directive prompt, when an open prompt?
  - When should I use user, system, or mixed initiative?

CS 224S W2006

## A threshold is a human-designed policy!

- Could we learn what the right action is
  - Rejection
  - Explicit confirmation
  - Implicit confirmation
  - No confirmation
- By learning a policy which,
  - given various information about the current state,
  - dynamically chooses the action which maximizes dialogue success

CS 224S W2006

## Another strategy decision

- Open versus directive prompts
- When to do mixed initiative

CS 224S W2006

## Review: Open vs. Directive Prompts

- Open prompt
  - System gives user very few constraints
  - User can respond how they please:
    - "How may I help you?" "How may I direct your call?"
- Directive prompt
  - Explicit instructs user how to respond
  - "Say yes if you accept the call; otherwise, say no"

CS 224S W2006

## Review: Restrictive vs. Non-restrictive grammars

- Restrictive grammar
  - Language model which strongly constrains the ASR system, based on dialogue state
- Non-restrictive grammar
  - Open language model which is not restricted to a particular dialogue state

CS 224S W2006 7

## Kinds of Initiative

- How do I decide which of these initiatives to use at each point in the dialogue?

Grammar	Open Prompt	Directive Prompt
Restrictive	<i>Doesn't make sense</i>	System Initiative
Non-restrictive	User Initiative	Mixed Initiative

CS 224S W2006 8

## Modeling a dialogue system as a probabilistic agent

- A conversational agent can be characterized by:
  - The current knowledge of the system
    - A set of states  $S$  the agent can be in
  - a set of actions  $A$  the agent can take
  - A goal  $G$ , which implies
    - A success metric that tells us how well the agent achieved its goal
    - A way of using this metric to create a strategy or policy  $\pi$  for what action to take in any particular state.

CS 224S W2006 9

## Goals are not enough

- Goal: user satisfaction
- OK, that's all very well, but
  - Many things influence user satisfaction
  - We don't know user satisfaction til after the dialogue is done
  - How do we know, state by state and action by action, what the agent should do?
- We need a more helpful metric that can apply to each state

CS 224S W2006 10

## Utility

- A utility function
  - maps a state or state sequence onto a real number
  - describing the goodness of that state
  - I.e. the resulting "happiness" of the agent
- Principle of Maximum Expected Utility:
  - A rational agent should choose an action that maximizes the agent's expected utility

CS 224S W2006 11

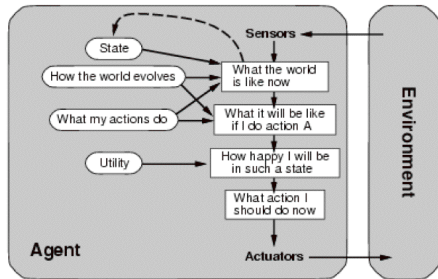
## Maximum Expected Utility

- Principle of Maximum Expected Utility:
  - A rational agent should choose an action that maximizes the agent's expected utility
- Action  $A$  has possible outcome states  $Result_i(A)$
- $E$ : agent's evidence about current state of world
- Before doing  $A$ , agent estimates prob of each outcome
  - $P(Result_i(A) | Do(A), E)$
- Thus can compute expected utility:

$$EU(A | E) = \sum_i P(Result_i(A) | Do(A), E) U(Result_i(A))$$

CS 224S W2006 12

## Utility (Russell and Norvig)



CS 224S W2006 14

## Markov Decision Processes

- Or MDP
- Characterized by:
  - a set of states  $S$  an agent can be in
  - a set of actions  $A$  the agent can take
  - A reward  $r(a,s)$  that the agent receives for taking an action in a state
  - (+ Some other things I'll come back to (gamma, state transition probabilities))

CS 224S W2006 14

## A brief tutorial example

- Levin et al (2000)
- A Day-and-Month dialogue system
- Goal: fill in a two-slot frame:
  - Month: November
  - Day: 12th
- Via the shortest possible interaction with user

CS 224S W2006 15

## What is a state?

- In principle, MDP state could include any possible information about dialogue
  - Complete dialogue history so far
- Usually use a much more limited set
  - Values of slots in current frame
  - Most recent question asked to user
  - Users most recent answer
  - ASR confidence
  - etc

CS 224S W2006 16

## State in the Day-and-Month example

- Values of the two slots day and month.
- Total:
  - 2 special initial and final states  $s_i$  and  $s_f$ .
  - 365 states with a day and month
  - 1 state for leap year
  - 12 states with a month but no day
  - 31 states with a day but no month
  - 411 total states

CS 224S W2006 17

## Actions in MDP models of dialogue

- Speech acts!
  - Ask a question
  - Explicit confirmation
  - Rejection
  - Give the user some database information
  - Tell the user their choices
- Do a database query

CS 224S W2006 18

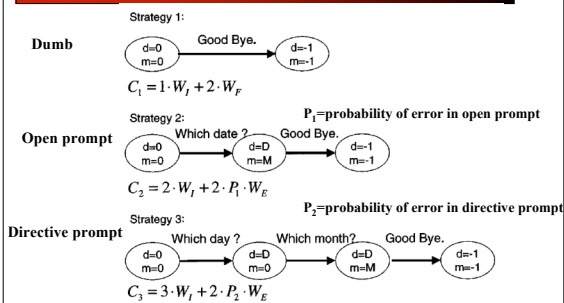
## Actions in the Day-and-Month example

- $a_d$ : a question asking for the day
- $a_m$ : a question asking for the month
- $a_{dm}$ : a question asking for the day+month
- $a_f$ : a final action submitting the form and terminating the dialogue

## A simple reward function

- For this example, let's use a cost function
- A cost function for entire dialogue
- Let
  - $N_i$ =number of interactions (duration of dialogue)
  - $N_e$ =number of errors in the obtained values (0-2)
  - $N_f$ =expected distance from goal
    - (0 for complete date, 1 if either data or month are missing, 2 if both missing)
- Then (weighted) cost is:
- $C = w_i \times N_i + w_e \times N_e + w_f \times N_f$

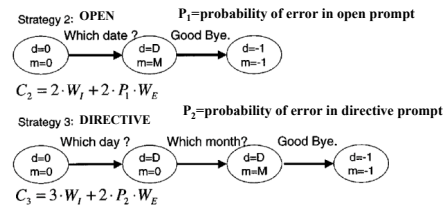
## 3 possible policies



## 3 possible policies

Strategy 3 is better than strategy 2 when improved error rate justifies longer interaction:

$$P_1 - P_2 > \frac{W_i}{2W_e}$$



## That was an easy optimization

- Only two actions, only tiny # of policies
- In general, number of actions, states, policies is quite large
- So finding optimal policy  $\pi^*$  is harder
- We need reinforcement learning
- Back to MDPs:

## MDP

- We can think of a dialogue as a trajectory in state space
- $$S_1 \rightarrow a_1, r_1 \ S_2 \rightarrow a_2, r_2 \ S_3 \rightarrow a_3, r_3 \ \dots$$
- The best policy  $\pi^*$  is the one with the greatest expected reward over all trajectories
  - How to compute a reward for a state sequence?

## Reward for a state sequence

- One common approach: discounted rewards
- Cumulative reward  $Q$  of a sequence is discounted sum of utilities of individual states

$$Q([s_0, a_0, s_1, a_1, s_2, a_2, \dots]) = R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots$$

- Discount factor  $\gamma$  between 0 and 1
- Makes agent care more about current than future rewards; the more future a reward, the more discounted its value

CS 224S W2006

25

## The Markov assumption

- MDP assumes that state transitions are Markovian

$$P(s_{t+1} | s_t, s_{t-1}, \dots, s_0, a_t, a_{t-1}, \dots, a_0) = P_T(s_{t+1} | s_t, a_t)$$

CS 224S W2006

26

## Expected reward for an action

- Expected cumulative reward  $Q(s,a)$  for taking a particular action from a particular state can be computed by Bellman equation:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

- Expected cumulative reward for a given state/action pair is:
  - immediate reward for current state
  - + expected discounted utility of all possible next states  $s'$
  - Weighted by probability of moving to that state  $s'$
  - And assuming once there we take optimal action  $a'$

CS 224S W2006

27

## What we need for Bellman equation

- A model of  $p(s'|s,a)$
- Estimate of  $R(s,a)$
- How to get these?
- If we had labeled training data
  - $P(s'|s,a) = C(s,s',a)/C(s,a)$
- If we knew the final reward for whole dialogue  $R(s_1, a_1, s_2, a_2, \dots, s_n)$
- Given these parameters, can use value iteration algorithm to learn  $Q$  values (pushing back reward values over state sequences) and hence best policy

CS 224S W2006

28

## Final reward

- What is the final reward for whole dialogue  $R(s_1, a_1, s_2, a_2, \dots, s_n)$ ?
- This is what our automatic evaluation metric PARADISE computes!
- The general goodness of a whole dialogue!!!!

CS 224S W2006

29

## How to estimate $p(s'|s,a)$ without labeled data

- Have random conversations with real people
  - Carefully hand-tune small number of states and policies
  - Then can build a dialogue system which explores state space by generating a few hundred random conversations with real humans
  - Set probabilities from this corpus
- Have random conversations with simulated people
  - Now you can have millions of conversations with simulated people
  - So you can have a slightly larger state space

CS 224S W2006

30

## An example

- Singh, S., D. Litman, M. Kearns, and M. Walker. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. Journal of AI Research.
- NJFun system, people asked questions about recreational activities in New Jersey
- Idea of paper: use reinforcement learning to make a small set of optimal policy decisions

CS 224S W2006

31

## Very small # of states and acts

- States: specified by values of 8 features
  - Which slot in frame is being worked on (1-4)
  - ASR confidence value (0-5)
  - How many times a current slot question had been asked
  - Restrictive vs. non-restrictive grammar
  - Result: 62 states
- Actions: each state only 2 possible actions
  - Asking questions: System versus user initiative
  - Receiving answers: explicit versus no confirmation.

CS 224S W2006

32

## Ran system with real users

- 311 conversations
- Simple binary reward function
  - 1 if completed task (finding museums, theater, winetasting in NJ area)
  - 0 if not
- System learned good dialogue strategy: Roughly
  - Start with user initiative
  - Backoff to mixed or system initiative when re-asking for an attribute
  - Confirm only a lower confidence values

CS 224S W2006

33

## State of the art

- Only a few such systems
  - From (former) ATT Laboratories researchers, now dispersed
  - And Cambridge UK lab
- Hot topics:
  - Partially observable MDPs (POMDPs)
  - We don't REALLY know the user's state (we only know what we THOUGHT the user said)
  - So need to take actions based on our BELIEF, i.e. a probability distribution over states rather than the "true state"

CS 224S W2006

34

## POMDP intuitions: Young et al 2005

- At each time step  $t$  machine in some hidden state  $s \in S$
- Since we don't observe  $s$ , we keep a distribution over states called a "belief state"  $b$
- So the probability of being in state  $s$  given belief state  $b$  is  $b(s)$ .
- Based on the current belief state  $b$ , the machine
  - selects an action  $a_m \in A_m$
  - Receives a reward  $r(s, a_m)$
  - Transitions to a new (hidden) state  $s'$ , where  $s'$  depends only on  $s$  and  $a_m$
- Machine then receives an observation  $o' \in O$ , which is dependent on  $s'$  and  $a_m$
- Belief distribution is then updated based on  $o'$  and  $a_m$ .

CS 224S W2006

35

## POMDP belief state update: Young et al (2005)

$$\begin{aligned} b'(s') &= P(s' | o', a_m, b) \\ &= \frac{P(o' | s', a_m, b) P(s' | a_m, b)}{P(o' | a_m, b)} \\ &= \frac{P(o' | s', a_m, b) \sum_{s \in S} P(s' | a_m, b, s) P(s | a_m, b)}{P(o' | a_m, b)} \\ &= \frac{P(o' | s', a_m) \sum_{s \in S} P(s' | a_m, s) b(s)}{P(o' | a_m, b)} \\ &= k \cdot P(o' | s', a_m) \sum_{s \in S} P(s' | a_m, s) b(s) \end{aligned}$$

CS 224S W2006

36

## POMDP reward

$$\begin{aligned}
 R &= \sum_{t=0}^{\infty} \lambda^t R(b_t, a_{m,t}) \\
 &= \sum_{t=0}^{\infty} \lambda^t \sum_{s \in S} b_t(s) r(s, a_{m,t}).
 \end{aligned}$$

CS 224S W2006

37

## How to learn policies?

- State space is now continuous
  - With smaller discrete state space, MDP could use dynamic programming; this doesn't work for POMDB
- Exact solutions only work for small spaces
- Need approximate solutions
- And simplifying assumptions

CS 224S W2006

38

## Young et al 2005 approach: factoring the state space

- Factor system state into 3 components:
- $S = b(s_u, a_u, s_d)$

$$\begin{aligned}
 b'(s'_u, a'_u, s'_d) &= k \cdot P(o'|a'_u) \sum_{s_u, a_u, s_d} P(s'_u|s_u, a_m) P(a'_u|s'_u, a_m) P(s'_d|s'_u, a'_u, s_d, a_m) b(s_u, a_u, s_d) \\
 &= k \cdot P(o'|a'_u) P(a'_u|s'_u, a_m) \sum_{s_u} P(s'_u|s_u, a_m) \sum_{s_d} P(s'_d|s'_u, a'_u, s_d, a_m) \sum_{a_u} b(s_u, a_u, s_d) \\
 &= k \cdot \underbrace{P(o'|a'_u)}_{\text{observation model}} \underbrace{P(a'_u|s'_u, a_m)}_{\text{user action model}} \sum_{s_u} \underbrace{P(s'_u|s_u, a_m)}_{\text{user goal model}} \sum_{s_d} \underbrace{P(s'_d|s'_u, a'_u, s_d, a_m)}_{\text{dialog model}} b(s_u, s_d)
 \end{aligned}$$

CS 224S W2006

39

## Other dialogue decisions

- Curtis and Philip's STAIR project
  - Do speaker ID
    - Yes: say "Hi, Dan"
    - probably: say "is that dan?"
    - Possibly: say "don't I know you?"
    - No: say "Hi, what's your name?"

CS 224S W2006

40

## Summary

- Utility-based conversational agents
  - Policy/strategy for:
    - Confirmation
    - Rejection
    - Open/directive prompts
    - Initiative
    - +?????
  - MDP
  - POMDP

CS 224S W2006

41

## Summary of Dialog in general

- The Linguistics of Conversation
- Basic Conversational Agents
  - ASR
  - NLU
  - Generation
  - Dialogue Manager
- Dialogue Manager Design
  - Finite State
  - Frame-based
  - Initiative: User, System, Mixed
- VoiceXML
- Information-State
  - Dialogue-Act Detection
  - Dialogue-Act Generation
- Evaluation
- Utility-based conversational agents
  - MDP, POMDP

CS 224S W2006

42